藏文紧缩格识别方法 *

拉玛扎西，才智杰†，扎西吉

(青海师范大学计算机学院，西宁 810008)

摘　要：分词是自然语言处理的一项基础性工作，对自然语言处理的后继工作有较大的影响。紧缩格的识别是藏文分词中最难最重要的技术之一。通过剖析已有藏文紧缩词识别方法，分析藏文字词的特征，针对性地提出了识别藏文紧缩格的规则算法、添加—还原算法和最大熵模型的特征模板，从而得到基于规则、添加还原法与最大熵模型相结合的藏文紧缩格识别方法。实验数据表明，该方法识别藏文紧缩格的准确率、召回率和 F1 值分别达 99.26%、96.47%、97.85%，比现有最高的准确率有了较明显的提高。

关键词：藏文；自然语言处理；分词；紧缩格

中图分类号：TP391.1　　doi: 10.3969/j.issn.1001-3695.2017.11.0747

Recognition method of Tibetan abbreviated case-auxiliary words

La Mazhaxi, Cai Zhijie†, Zha Xiji

(*Computer School of Qinghai Normal University*, *Xining* 810008, *China*)

Abstract: Word segmentation is a basic work of natural language processing, which has a great influence on the subsequent work of it, the recognition of abbreviated case-auxiliary words is one of the most difficult and important technologies of Tibetan word segmentation. Through dissecting the existing recognition methods of abbreviated case-auxiliary words, this paper analyzed the characteristics of Tibetan words, targetedly proposed recognition algorithm of Tibetan abbreviated case-auxiliary words rules, add - restore algorithm and the maximum entropy models feature template, then the methods of recognizing abbreviated case-auxiliary words based on the rules, add-restore methods and the maximum entropy model were obtained. The experimental data showed that the accuracy, recall rate and F value of the method is 99.26%, 96.47%, and 97.85% respectively , which shows a obvious progress than that of the existing methods.

Key words: Tibetan; NLP; segmentation; abbreviated case-auxiliary words

0　引言

藏文是一种典型的逻辑格语法体系的复杂拼音文字 [1]，由实词和虚词按一定的语法结构组合而成。计算机正确识别虚词对文本的歧义消解和句法、句型、语义处理有着重要的意义，虚词中的 la 格助词“ར”、具格助词“ས”、属格助词“འི”、终结词“འོ”、饰集词“འང”和离合词“འམ”与其前一音节不加分字分隔符组成一个音节，在藏语自然语言处理中称这些虚词为紧缩词。紧缩词的识别既是藏文分词的一项基础工作，也是藏文分词的难点，为此学者们围绕紧缩词的识别展开了研究。紧缩词中属格助词“འི”、终结词“འོ”、饰集词“འང”和离合词“འམ”的识别基本得到解决，由于 la 格助词“ར”和具格助词“ས”的情况比较复杂，目前的识别方法还有待改进。本文在现有紧缩词识别方法的基础上，提出了一种规则、还原法和最大熵相结合的 la 格助词“ར”和具格助词“ས”（下文称此为紧缩格）识别的混合策略。

1　研究现状

自 1999 年学者们开始研究藏文分词问题以来，取得了很多有价值的成果。在藏文分词方面，陈玉忠等人[2]首次提出了一种基于格助词和接续特征（BCCF）的书面藏文自动分词方案；这一方案消除了切分歧义和未登录词识别问题，提高了藏文分词精度，其最终切分准确率达 97.21%。才智杰[3]设计开发了班智达藏文自动分词系统。该系统采用的是基于词典匹配的分词

收稿日期: 2017-11-22; **修回日期:** 2018-01-30　**基金项目:** 国家自然科学基金项目(61163018, 61262051); 国家社科基金项目(13BYY141, 16BYY167); 教育部“春晖计划”项目（Z2012093，Z2016077）；青海省基础研究计划项目（2017-ZJ-767）；“长江学者和创新团队发展计划”创新团队项目（IRT1068）；藏文信息处理与机器翻译重点实验室（2013-Y-17）

作者简介: 拉玛扎西（1994-），男，藏族，甘肃夏河人，硕士研究生，主要研究方向为藏文信息处理，藏语自然语言处理（lhamatashi@outlook.com）；才智杰（1970-），男（通信作者），藏族，青海乐都人，教授、硕导，博士研究生，主要研究方向为藏文信息处理，藏语自然语言处理；扎西吉（1992-），女，藏族，甘肃夏河人，硕士研究生，主要研究方向为藏文信息处理，藏语自然语言处理.

方法，在 85 万字节藏语语料的切分准确率达 99%。刘汇丹等人[4]采用格助词分块并识别临界词，采用最大匹配方法分词，系统最终分词正确率达 96.98%。史晓东等人[5]将基于 HMM 的汉语分词系统 Segtag 移植到藏文分词中，设计实现了央金藏文分词系统，其准确率达 91%。康才畯[6]在常用的四词位标注集扩充为六词位标注集，采用条件随机场作为标注建模工具来进行训练和测试，准确率达 95.89%。龙从军等人[7]用 CRF 六字位分词，准确率达 94.34%。李亚超等人[8]基于条件随机场模型实现了基于音节标注的藏文分词系统，准确率达 95.35%。洛桑嘎登等人[9]采用条件随机场和规则融合方法解决藏文分词问题，最终正确率为 96.11%。李亚超等人[10]从无标注语料中抽取边界熵特征、邻接变化数特征、无监督间隔标注等无监督特征，并将融合到基于序列标注的分词系统中，其分词 F 值提高了 0.97%。由以上文献可以看到，藏文分词主要采用规则法[2~4]、统计法[5~8]和规则与统计相结合[9,10]等三种。在这三种方法中，规则法适合于封闭语料的切分，开放语料下准确率有所下降，主要存在的问题是无法识别未登录词和命名题识别问题。统计法通过对语料的训练，自动分析文本特征，从而达到文本的分词问题，能够弥补规则法的不足；但统计法需要大规模的分词语料做支撑，并且要选择适合的语言模型。由于现今的藏文分词语料规模较小，分词语料的准确率也较低，所以只采用统计法还不乐观。整体来看采用多种分词法相结合比较合适目前的藏文分词。

藏文分词中有一类特殊的词称紧缩词，紧缩在藏文中所占的比较很大，因此藏文紧缩词的识别是藏文分词必须要解决的问题。陈玉忠等人[2]在藏文分词难点分析中指出，500 句藏文综合语料中切分错误率占总词数的 12.71%，其中紧缩词的切分错误占 6.93%。说明紧缩词的识别是藏文分词的难点，学者们围绕藏文紧缩词的识别展开了研究。才智杰[11]系统地阐述了紧缩词在藏文信息处理中的核心地位，提出了紧缩词的“添加—还原法”识别方法，在 85 万字节的语料中测试，紧缩词的识别准确率达 99.83%，取得了较好的效果。完么扎西等人[12]在“添加—还原法”的基础上利用藏文文法约束规则识别紧缩词，在含有 4 040 个紧缩词的文本中识别准确率达 99.95%。李亚超等人[13]再次分析了紧缩词在藏语分词中的地位，指出文献[11,12]中的方法需要词库支持，无法识别未登录词后的紧缩词。为解决这个问题，提出了基于条件随机场的紧缩词识别方法，其识别准确率达 98.91%。虽然此方法的准确率比基于规则的准确率低，但在一定程度上克服了“还原法”中不能识别“未登录词+紧缩词”的问题。华却才让等人[14]在基于音节特征感知机训练模型的藏文命名实体识别方案中，重点研究了利用藏文紧缩词识别音节的方法，其识别准确率达到了 99.91%。康才畯等人[15]采用基于词位的统计分析方法识别并切分现代藏语文本中的藏文紧缩词，准确率为 95.89%，其最大特点是减少了未登录对识别效果的影响。以上文献采用规则法和统计法研究了藏文紧缩词的识别方法，龙从军等人[9]采用统计和规则相结合的方法研究了紧缩词的识别问题，其准确率达 98.01%。从以上研究情况来看，规则法适合于封闭语料下紧缩词的识别，统计法不受语料的限制，合适于开放语料中紧缩词识别，规则法和统计法相结合既不全依赖词库，又不完全受训练语料的质量的影响，是藏文紧缩词识别比较有效的方法。紧缩词识别准确率对比见表 1。

表 1　紧缩词识别准确率对比

	方法	测试语料	紧缩格识别准确率%		接续紧缩词识别准确率%			平均准确率%
			འི	ར 和 ས	འམ	འང	འོ	
规则	添加-还原法[11]	中小学教材	100	99.15	100	100	100	99.83
	文献[12]	综合 1[1]	-	-	-	-	-	99.95
统计	条件随机场[13]	政论	98.48	98.88	100	100	98.71	98.91
	感知机[14]	综合 2[1]	-	-	-	-	-	99.91
	CRF++[15]		-	-	-	-	-	93.20
	黏写分词一体化切分[7]	新闻 中小学教材	94.80	94.30	53.33	77.32	-	82.81
	双标签黏写切分[7]		95.20	92.81	53.33	84.04	-	83.64
	五标签黏写切分[7]		94.19	91.65	53.33	75.26	-	81.22
	规则和统计相结合[7]		100	98.79	96.67	95.88	-	98.01

由表 1 可见，紧缩词“འི་འམ་འང་འོ”可直接用规则的方法识别，其准确率达 100%；紧缩格“ར་ས”用规则法识别率相对较低，尤其多种类型的语料中其识别率明显下降。而统计法能克服“未登录词+紧缩词”的现象，在各种统计方法中，识别紧缩格的最高准确率达 98.88%，但对使用频率极高的紧缩格来说还不能满足实际需求。

2　紧缩格识别

2.1　紧缩格的特征

为了便于叙述本文把还未判定是否为紧缩格的“ར་ས”称为拟紧缩格，含拟紧缩格的音节称为拟紧缩音节。紧缩格“ར་ས”识别的难点究其原因有以下几点：

a) 紧缩格只能出现在后加字位置，即出现在后加字位置上

的“ར་ས”为拟紧缩格。

b) 紧缩格与后加字兼类。例如，文本“ཟི་ལིང་ནས་ལྷ་ས་བར་རྐང་ཐང་ལ་སོང་། ཟི་ལིང་ནས་ལྷ་ས་བར་མགྲོན་བྱས།（从西宁步行到拉萨，在西宁请拉萨人）”中第一个拟紧缩音节“བར”后的“ར”是后加字，第二个拟紧缩音节“བར”中的“ར”是紧缩格；又如文本“རི་བོ་གདང་ལ་ལས་བབས་པའི་ཆུ། ལ་ལས་རློག་བྱ་སྐྱུར་བཞིན་ཡོད།（源自唐古拉山的水，有些人在放风筝）”中第一个拟紧缩音节“ལས”后的“ས”是后加字，第二个拟紧缩音节“ལས”后的“ས”是紧缩格。

c) 紧缩格不能与其他格助词重叠接续。吉太加[16]指出，藏文格助词不能重叠使用。例如“ཐུག་ཐུག་རིག་རིག་ཚང་མ་བདག་གིར་བྱས་པའི་བསམ་པ།（见到什么都想己有的想法）”的“གིར”中的拟紧缩格“ར”不是紧缩格。若其识别为紧缩格，对应的还原结果为“'''གི་རུ'''”，这就使句中出现了两个格助词重叠接续的现象。

d) 识别时需考虑上下文语境，紧缩格具有动态性，同一个拟紧缩格在不同的语境中有不同的识别结果。例如，单音节“བར”、“པར”、“ཆས”、“རས”等具有实际意义，可单独成词；但在有些语句中这些音节后的“ར”、“ས”是紧缩格，因此识别紧缩格时需具体分析上下文的语境。例如，“རང་གི་ལ་རྒྱ་ཕྱིར་རྩོད་ཕྱིར་དུ་ཚང་མ་ལ་གསལ་ཁ་ཕྱེས།（为了赢得尊严说明实况）”中两个相同的拟紧缩格音节“ཕྱིར”有不同的识别结果，其第一个“ར”为紧缩格，第二个不是紧缩格。

2.2 紧缩格识别

2.2.1 紧缩格的规则及添加还原识别算法

本文采用模块式分步方案识别紧缩格，由文本读取、规则法识别、还原法识别、最大熵模型识别和文本输出五个模块组成。读取文本后，首先提取拟紧缩音节，对其用规则识别。若规则不能识别，则用还原法识别；若还原法不能识别，则用最大熵模型识别。其识别过程如图 1 所示。

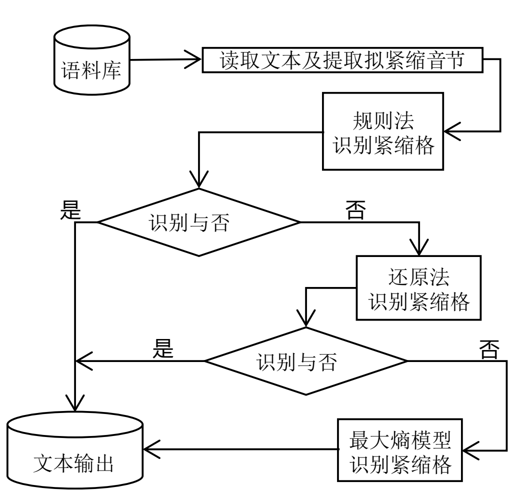


图 1　紧缩格识别模型

文本读取后先定位拟紧缩音节的位置 i,并读取其前后各 2 个音节作为待处理的五元处理对象，即处理对象 $w=<w_{i-2},w_{i-1},w_i,w_{i+1},w_{i+2}>$。当拟紧缩音节 w_i 前或后不足 2 个音节时，前面的不足位补充“start”，后面的不足位补充“end”。选择五元处理对象的原因是：一方面控制算法时间复杂度，另一方面藏文 1~4 音节词占总词条数的 93.78%，4 音节词以上的只占总词条数的 6.22%[17]。如文本“ངས་སྒྲོལ་མར་ལས་བྱ་འབྲི་རོགས་བྱས།（我帮卓玛写作业）”中，对拟紧缩音节“ངས”、“མར”、“ལས”和“བྱས”等提取五元处理对象<start，start，ངས,སྒྲོལ,མ>、<ངས,སྒྲོལ,མར,ལས,བྱ>、<སྒྲོལ,མར,ལས,བྱ,འབྲི>和<འབྲི,རོགས,བྱས，end，end>。拟紧缩音节识别算法、紧缩格的规则识别算法及添加—还原算法如下：

算法 1　拟紧缩音节识别算法

```
输入：藏文字 s
输出：1 或 0，1 表示 s 是拟紧缩音节，0 表示 s 是非紧缩音节
if(length(s)>1 且尾字符为‘ས’或‘ར’)
   if(length(s)>=3 且尾字符前一字符为‘ག་ང་བ་མ’之一)
     if(length(s)=3 且尾字符之前字符为‘དག་དང་དབ་དམ་འག་འབ་མག’
之一) //除去了再后加字为ས的情况
            return   1;
     else   return   0;
    else  return  1;
else  return  0;
```

算法 2　紧缩格的规则识别算法

```
输入：w=<wi-2,wi-1,wi,wi+1,wi+2>;//w 为拟紧缩格的五元处理对象
输出：1 或 0 或-1，1 表示拟紧缩格音节 wi 为紧缩音节，0 表示非紧
缩音节，-1 表示无法判断
if(wi 为‘གིས་ཀྱིས་གྱིས་འིས་ཡིས’之一)
     return   0;
else   if(wi 为‘ངས་འདིས་དེར་འདིར’之一)
             return    1;
        else  if(wi-1=‘ཨ’)
                if((wi-1+为‘ཨ་མྱེས།ཨ་གར།ཨ་ཀར།ཨ་གསར།
ཨ་སྒོར།ཨ་འཐས།ཨ་ཟེར།ཨ་བཙས།ཨ་བར།’之一)
                     return   0;
                else   return    1;
        else  if(wi or wi-1+wi or  wi-2+wi-1+wi  or  wi+wi+1
or  wi+wi+1+wi+2∈DB1)
                return   0;
           else  return    -1;
```

“ངས་འདིས་དེར་འདིར”等 4 个字出现的频率非常高，其中的“ས 或 ར”在任何情况下都为紧缩格，因而用规则识别较为合适；二音节词中第一个音节为“ཨ”，第二个音节的最后一个字符为“ས 或 ར”的词“ཨ་མྱེས། ཨ་གར། ཨ་ཀར། ཨ་གསར། ཨ་སྒོར། ཨ་འཐས། ཨ་ཟེར། ཨ་བཙས། ཨ་བར།”等中的“ས 或 ར”不是紧缩格，其他都是紧缩格，这种情况也可以用规则法识别。DB1 存放了单音节、双音节和三音节字中最后一个字符在任何情况下都为后加字的词，这类词共有 1622 个，如“ཤེས། ཆོས། གསོ་བཅོས། འགྲུལ་རིས། ལས་གཞི། ཟུར་མིག། སློབ་གསོ་ཚུས། བདག་གིར་བཞེས། མར་ཁེ་སི། དམར་པོ་རི།”等。例如，“བཀྲ་ཤིས་ཀྱིས་ལས་བྱ་འབྲི་ཞོར་དུ་ཨ་ཕར་ཁ་པར་བཏང་སྟེ་ཕ་ཡུལ་གྱི་གནས་ཚུལ་དྲིས།（扎西在写作业的同时给爸爸打电话问家里情况）”，

通过算法 1 提取例句中的拟紧缩音节“ཞིས、ཀྱིས、ལས、ཞོར、ཕར、པར、གནས、དྲིས”，由算法 2 可识别“ཀྱིས、ལས、པར、གནས、དྲིས”中的‘ས 或 ར’为后加字，“ཕར”中的‘ར’是紧缩格，由紧缩格的特征（3）可以判断“ཞོར”中的‘ར’是后加字，规则法无法判断“ཞིས”中的‘ས’是否为紧缩格。

算法 3 紧缩格的“添加-还原” 算法

```
输入：w=<wi-2,wi-1,wi,wi+1,wi+2>;//w 为拟紧缩格的五元处理对象
输出：1 或-1，1 表示拟紧缩音节 wi 为紧缩音节，-1 表示无法判断
w’= wi- ‘拟紧缩格’；
w’’=wi-1+wi- ‘拟紧缩格’；
w’’’=wi-2+wi-1+wi- ‘拟紧缩格’；
if(w’中最后一个字符或者倒数第二个字符为下加字或元音或上加字)
    if(w’’或者 w’’’∈DB2)
        return  1；
    else  return -1；
else   if(w’’+ ‘འ’ 或者 w’’’+ ‘འ’ ∈DB2)
        return  1；
    else  return  -1；
```

DB2 中存放了二音节词、三音节词且最后一个音节是无后加字或后加字为‘འ’的词，主要用于利用“添加-还原”法判断紧缩格，如“ཇོ་ཇེ། མཁན་པོ། ནམ་མཁའ། པ་མཐའ། རིན་པོ་ཆེ། ག་གེ་མོ། སའི་གོ་ལ།”等词。

2.2.2 紧缩格的最大熵识别的方法

Jaynes 于 1957 年首次提出最大熵原理之后，被广泛应用于自然语言处理领域。其基本原理为：在已知部分信息的前提下，关于未知分布最合理的推断应该是符合已知信息最不确定或最大随机的推断[18]。藏文紧缩格识别可看做是一个序列标注问题，标注时对每个对象随机标注一个标签，并建立已知特征 x 的条件下输出标签 y 的概率分布模型 p(p∈P)。其中 x 属于上下文信息集 X(x∈X)，而 y 属于对应的标签集 Y(y∈Y)。从训练集中可获得 N 个样本集，即 S={(x1,y1),(x2,y2),⋯,(xn,yn)}，根据这些样本可以定义一个事件空间，其特征是一个二值函数 f:X×Y→{0,1}，其定义如下：

$$f(x,y)=\begin{cases}1 & \text{if } x=x_i \text{ and } y=y_i,(x_i,y_i)\\ 0 & \text{otherwise}\end{cases}$$

则模型 p 的熵为：

$$H(p)=-\sum_{x,y}p(x,y)\log\left(p(x,y)\right) \tag{1}$$

从式(1)中可得出最大熵模型为

$$P^{*}=\arg\max_{p\in c}H(p) \tag{2}$$

式(2)中的 C 为符合约束条件的模型集合，然后计算满足 C 条件的最大 p*：

$$P^{*}(y\mid x)=\frac{1}{z(x)}\exp\left(\sum_i\lambda_i f_i(x,y)\right) \tag{3}$$

其中：Z(x)是归一化常数，并有

$$z(x)=\sum_y\exp\left(\sum_i\lambda_i f_i(x,y)\right) \tag{4}$$

式(3) (4) 中的 λ_i 为模型参数，即特征 f_i 对应的权重 λ_i，可通过 IIS 算法[14]来估计。

藏文紧缩格的识别需要把原始文本内容进行序列标注，如“གནས་འདིར་ཟམ་པ་མེད་པས།(因这里没有桥)”的文本语料，经过标注后的训练语料为“གནས་F་འདིར་Tཟམ་N་པ་N་མེད་N་པས་T།”。其中 T 表示紧缩格，F 表示后加字，N 表示不是拟紧缩格。首先将{T，F，N}作为标签集，每个音节及其上下文信息作为输入值；然后使用最大似然估计统计语料中每个特征概率；最后选取模型返回的每个标签的最大输出概率。

最大熵模型中针对研究对象选择有效的上下文特征是一个关键问题，根据藏文词语音节的分布特点及上下文激发环境确定了模型，并抽取特征模板。本文选取的特征模板见表 2。

表 2 特征模板

序号	原子模板	模板意义
1	NJ	当前字（拟紧缩格）
2	LNJ	拟紧缩格和其前一音节
3	RNJ	拟紧缩格和其后一音节
4	NJL	拟紧缩格和其后两个音节
5	NJR	拟紧缩格和其前两个音节
6	NJR_R	去掉紧缩格的字串和其前一音节

训练语料的每个音节都采用以上特征模板，拟紧缩格前的音节不足时，用“start”补充，拟紧缩格后音节不足时，用“end”补充，统计模型采用最大熵模型开源程序包。

3 实验数据

本文在青海师范大学建立的语料中选取了含 66 184 个字的语料作为测试语料（其中拟紧缩格有 9 387 个），对已有紧缩格识别方法和本文提出的紧缩格识别方法进行了测试。为了确保语料的准确性，全部经过人工反复校对。语料领域包括政治、教材、历史、小说、新闻五种题材，测试方式包括开放测试和封闭测试两种。封闭测试时，用全语料进行训练，然后随机选取 30%做测试语料；开放测试时，其中的 80%做训练语料，20%做测试语料。实验结果见表 3。其中，方法 A 指文献[4]采用的规则、统计相结合的方法，方法 B 是文献[11] 提出的添加还原法，方法 C 是文献[13] 采用的条件随机场技术，方法 D 是文献[14]采用的感知机技术；方法 E1、E2、E3、E4 分别表示本文提出的紧缩格的规则识别方法、“添加—还原” 算法、最大熵识别的方法、“规则+还原法+最大熵”法（简称为混合法）。

由表 3 可见，紧缩格的识别仅用规则法或统计法其效果不佳，规则和统计相结合的方法识别紧缩格的准确率较高。本文

结合规则、添加还原法和最大熵三种方法识别藏文紧缩格，在封闭语料上的测试准确率、召回率和 F1 值分别达 99.81%、99.19% 和 99.50，在开放语料上的测试准确率、召回率和 F1 值分别达 99.26%、96.47%和 97.85，比现有最高的准确率有了较明显的提高。

表 3　紧缩格识别实验数据

方法	测试	准确率/%	召回率/%	F1 值/%
A	开放	98.19	91.50	94.73
B	开放	95.51	78.96	86.44
C	开放	95.92	78.95	86.61
D	开放	94.11	66.60	78.00
E1	开放	85.29	85.29	85.29
E2	开放	42.70	23.03	29.92
E3	开放	95.74	81.28	87.92
E4	封闭	99.81	99.19	99.50
	开放	99.26	96.47	97.85

在方法 E4 中主要出现了两类错误，一类是拟紧缩格后的格助词没能正确识别，例如，“ཤིང་བཟོ་ཀུན་དགས་དུ་བའི་རྟ་ཞོན་ནས་ནམ་མཁར་འཕྱུར།（木匠更嘎乘飞烟在空中遨游）”中，由于把兼类格助词“དུ”识别成了格助词，从而没能识别拟紧缩音节“དགས”中的紧缩格，这类错误可以通过提高格助词识别得以解决；另一类是训练语料没能覆盖，这类错误可通过增大训练语料的方式弥补。

4　结束语

藏文紧缩格的识别是藏文分词中最难最重要的技术之一。本文通过剖析已有藏文紧缩词识别方法，分析藏文紧缩格的特征，针对性地设计了识别藏文紧缩格规则算法、添加—还原算法和最大熵模型的特征模板，结合三种算法对藏文紧缩格进行了识别。实验数据表明，该方法识别藏文紧缩格的准确率、召回率和 F1 值分别达 99.26%、96.47%、97.85%，比现有最高的准确率有了较明显的提高。今后在此基础上研究藏文分词技术，使藏文分词尽早满足实际需求。

致谢：本文在测试紧缩格的识别效果时，中国科学院刘汇丹老师、西北民族大学李亚超老师、青海师范大学华却才让老师和完么扎西老师等给予了帮助，特此表示衷心的感谢！

参考文献：

[1] 孙萌，刘群. 基于判别式分类和重排序技术的藏文分词 [C]// 第十二届全国少数民族语言文字信息处理学术研讨会论文集. 2011.
[2] 陈玉忠，李保利，俞士汶. 藏文自动分词系统的设计与实现 [J]. 中文信息学报, 2003, 17 (3): 15-20.
[3] 才智杰. 班智达藏文自动分词系统的设计与实现 [J]. 青海师范大学民族师范学院学报, 2010, 21 (2): 75-77.
[4] 刘汇丹，诺明华，赵维纳，等. SegT: 一个实用的藏文分词系统 [J]. 中文信息学报, 2012, 26 (1): 97-103.
[5] 史晓东，卢亚军. 央金藏文分词系统 [J]. 中文信息学报, 2011, 25 (4): 54-56.
[6] 康才畯. 藏语分词与词性标注研究 [D]. 上海：上海师范大学, 2014.
[7] 龙从军，刘汇丹. 藏文自动分词的理论与方法研究 [M]. 北京：知识产权出版社, 2016.
[8] 李亚超，江静，加羊吉，等. TIP-LAS: 一个开源的藏文分词词性标注系统 [J]. 中文信息学报, 2015, 29 (6): 204-207.
[9] 洛桑嘎登，杨媛媛，赵小兵. 基于知识融合的 CRFs 藏文分词系统 [J]. 中文信息学报, 2015, 29 (6): 213-219.
[10] 李亚超，加羊吉，江静，等. 融合无监督特征的藏文分词方法研究 [J]. 中文信息学报, 2017, 31 (2): 72-75.
[11] 才智杰. 藏文自动分词系统中紧缩词的识别 [J]. 中文信息学报, 2009, 23 (1): 35-37.
[12] 完么扎西，尼玛扎西. 藏语自动分词中的几个关键问题的研究 [J]. 中文信息学报, 2014, 28 (4): 132-139.
[13] 李亚超，加羊吉，宗成庆，等. 基于条件随机场的藏语自动分词方法研究与实现 [J]. 中文信息学报, 2013, 27 (4): 51-58.
[14] 华却才让，姜文斌，赵海兴，等. 基于感知机模型藏文命名实体识别 [J]. 计算机工程与应用, 2014, 50 (15): 172-176.
[15] 康才畯，龙从军，江荻. 基于词位的藏文黏写形式的切分 [J]. 计算机工程与应用, 2014, 50 (11): 218-222.
[16] 吉太加. 藏文语法研究 [M]. 青海：青海民族出版社, 2013.
[17] 才智杰，才让卓玛. 班智达藏文标注词典设计 [J]. 中文信息学报, 2010, 24 (5): 46-49.
[18] 宗成庆. 统计自然语言出处理 [M]. 2 版. 北京：清华大学出版社, 2013, 81-128.
[19] Liu Huidan, Zhao Weina, Nuo Minghua, *et al.* Tibetan word segmentationas syllable-tagging using conditional random fields [C]// Proc of the 25th Pacific Asia Conference on Language, Information and Computation. 2011: 168-177.
[20] 于江德，王希杰，樊孝忠. 基于最大熵模型的词位标注汉语分词 [J]. 郑州大学学报：理学版, 2011, 43 (1): 70-74.